DOSKEY Macro Tips

BY NEIL J. RUBENKING

CREATING A DVORAK KEYBOARD

DVORAK.SCR (Figure 1) produces a memory-resident utility requiring only 112 bytes of RAM. When installed, the utility enables the Dvorak keyboard layout shown in Figure 2. When Scroll Lock is OFF, the Dvorak layout is active. When Scroll Lock is ON, the QWERTY layout is active.

DVORAK.COM makes use of interrupt 15h, service 4Fh, so computers with a BIOS older than January 1986 may not be able to use it. The program is intended for use with DOS; Windows already includes a Dvorak keyboard layout.

In all, 33 keys are translated. Note that there are minor Dvorak variants dealing with the quote, slash, and semicolon keys; you may find other Dvorak layouts that place these keys differently. Key redefinition programs that use interrupt 16h can be used with this program in their normal fashion.

Create DVORAK.SCR with an ASCII editor. Be sure to put a carriage return after the final Q and a blank line before the R CX line. To create the program, enter `DEBUG < DVORAK.SCR` at the DOS prompt, or you can download DVORAK.ZIP from the Utilities/Tips Forum on PC MagNet.

David M. Dibble
Riverside, California

▶PC MAGAZINE: There's a persistent rumor that the original purpose of the standard QWERTY keyboard layout was to prevent jammed keys by slowing down typists. Whether or not this is true, Dvorak aficionados insist that it speeds their typing by putting the most-used keys on the home row. DVORAK.COM lets you try this layout for yourself.

The author warns that computers with a BIOS date before January 1986 may not be able to use DVORAK.COM. If you don't know the date on your computer's BIOS, you can find out easily enough. Load DEBUG and enter the following command: `D FFFF:5 L 8`. The resulting line of information will include the BIOS date at the far right. Type Q and Enter to quit DEBUG.

DVORAK.SCR is a rather long DEBUG script, and presents many opportunities for error. You can catch any errors by creating DVORAK.COM with this command:

```
DEBUG < DVORAK.SCR > DVORAK.LOG
```

Now scan the .LOG file for the string "Error. If the string appears, there was an error in the line above; fix it and try again. Only when DEBUG processes the script with no errors should you run the DVORAK program.

DVORAK.COM saves memory by disposing of its own copy of the DOS environment. But if you check your memory with DOS 5.0's MEM command or another memory-mapping utility, you'll see that DVORAK.COM takes 368 bytes (170 in hexadecimal), not 112. It's true that the program code takes only 112

DVORAK.SCR

Complete Listing

```
N DVORAK.COM
A 100
JMP 0168                    ;Jump to INITIALIZE
PUSHF                       ;**BEGIN  replace int 15h
CMP AH, 4F                  ;called by Int 9 to translate scan codes
JZ 0112                     ;if function 4Fh, jump TRANSLATE
POPF                        ;restore flags
JMP 010D                    ;jump ORIGINAL handler
POP BX                      ;DO_INT15
STC
JMP 0000:0000               ;ORIGINAL int 15h
POPF                        ;TRANSLATE
PUSH BX
PUSH DS
MOV BX, 0040
MOV DS, BX                  ;DS to BIOS data area
TEST BYTE PTR [0017],10     ;Scroll Lock on?
POP DS
JNZ 010B                    ;if on, use Qwerty keyboard.  Jump DO_INT15
MOV AH, AL                  ;scan code to AH
AND AH, 80                  ;save possible high bit, AH = 80h or 0
AND AL, 7F                  ;zero high bit
CMP AL, 0C                  ;scan code in range?
JB  0138                    ;  if not, jump BREAK
CMP AL, 35
JA  0138                    ;Jump if high range to BREAK
SUB AL, 0C                  ;set up AL as pointer
MOV BX, 013E                ;BX to address of KEY_LIST scan code table
CS:                         ;       key values 12 - 53 (decimal)
XLAT                        ;translate AL to new scan code. Key pressed.
OR AL, AH                   ;if BREAK, OR with 80h.  Key released
MOV AH, 4F                  ;restore function 4Fh
JMP 010B                    ;Jump DO_INT15.  KEY_LIST follows:
DB          1A  1B  0E  0F  28  33  34  19
DB  15  21  22  2E  13  26  35  0D  1C  1D
DB  1E  18  12  16  17  20  23  14  31  1F
DB  0C  29  2A  2B  27  10  24  25  2D  30
DB  32  11  2F  2C
MOV AX, 3515                ;INITIALIZE.  Above portion is TSR
INT 21
MOV [010E], BX              ;save original int 15h vectors
MOV [0110], ES
MOV AX, 2515                ;code to set interrupt
MOV DX, 0102                ;replacement int 15h address
INT 21
MOV AX, [002C]              ;environment block
MOV ES, AX
MOV AH, 49                  ;free our copy of environment
INT 21
MOV DX, 0192                ;load DX with SIGN_ON
MOV AH, 09                  ;display it
INT 21
MOV DX, 0168                ;DX to INITIALIZE, area to place in memory
INT 27                      ;go memory resident.  SIGN_ON follows:
DB  0D,0A,'DVORAK loaded.',0D,0A,24

R CX
A5
W
Q
```

Figure 1: **This DEBUG script generates a minuscule utility program that lets you choose between the regular QWERTY keyboard layout and the Dvorak layout.**

Figure 2: **The Dvorak keyboard layout supposedly promotes faster typing by putting the most often used keys on the home row.**

bytes, but the obligatory Program Segment Prefix adds another 256. Even at 368 bytes, it's amazingly tiny for what it does!

DOSKEY MACRO TIPS

The DOS 5.0 reference manual describes only a few of the tricks you can perform using DOSKEY macros. The manual indicates three ways, for instance, to execute ANSI escape sequences:

- using the PROMPT command
- using TYPE to display a text file containing the ANSI sequences
- using ECHO in a batch file

But it never suggests you can use a DOSKEY macro to execute ANSI escape sequences directly from the keyboard:

```
DOSKEY ESC=ECHO ^[[$*
```

Here, ^[represents the escape character. To use the macro, type ESC followed by any ANSI codes at the DOS prompt. Once you've installed the macro, for example, you could enter the following command to program the Home key to change to the root directory of drive C: and clear the screen

```
ESC 0;71;"c:";13;"
 cd\";13;"cls";13p
```

A variation of this lets you send escape codes to control your printer directly from the DOS prompt. First create the PESC macro by entering the line

```
DOSKEY PESC=ECHO
 ^[$*$ GPRN
```

Then you can add the appropriate characters to accomplish your task. For example, the command

```
PESC4^N This is a test
```

would print "This is a test" in double-wide italics on a printer that uses Esc 4 for italics and Ctrl-N for double-wide printing.

By omitting the escape character from this macro definition you can also define a line-printing macro that will send anything you type after LP to the printer:

```
DOSKEY LP=ECHO $* $G PRN
```

Since you can't enter the escape character directly from the keyboard, you'll need to use a text editor to create a batch file that defines the macro. To enter an escape or other control character using DOS 5.0's EDIT.COM, first press Ctrl-P and then the special key. The escape character appears as a left arrow. Some control characters, like Ctrl-L, are intercepted by the DOS editor. To enter these, press Ctrl-P followed by Alt-*nn*, where *nn* is the ASCII code for the character. The following useful macro will eject a page from the printer when you type FF:

```
DOSKEY FF=ECHO ^L $G PRN
```

To enter ^L using the DOS editor, type Ctrl-P, then Alt-12.

Finally, if you use a lot of macros and can't remember them all, just add the following ANSI command to your macros batch file so you can press Page Down to see a list of the available macro commands:

```
ECHO^[[0;81;"DOSKEY
 /m¦sort¦more";13p
```

Here again, the ^[after ECHO represents the escape character.

Brian L. Zimmerman
El Paso, Texas

▶PC MAGAZINE DOSKEY macros are a little like batch files, but they have their own special abilities and limitations. The biggest limitation is that the entire macro must fit on a single 127-character command line, though within that line the $T special character can separate multiple commands. The strongest advantage is that DOSKEY macros are stored in memory, so they execute much faster than batch files. Also, even the smallest batch file takes one cluster (2K or more) of disk space, yet you can probably create all of your DOSKEY macros with a single batch file that's under 2K.

Another virtue of DOSKEY macros, one that I wish were available to batch files, is their ability to access the command line as a whole. Batch files can access individual command line elements using the replaceable parameters %1 through %9. DOSKEY macros use $1 through $9 for the same purpose, but the special character $* is replaced by *everything* that follows the macro name. The ESC, PESC, and LP macros above all use this special character. Note also that PESC and LP include the special character $G, which is replaced by the greater-than symbol when the macro runs.

The May 26, 1992, User-to-User column presented a BASIC program called SAVEMAC.BAS. Used in combination with a DOSKEY macro of the same name, this program creates LOADMAC.BAS, a batch file that recreates all active DOSKEY macros. When you're experimenting with various DOSKEY macros, it's handy to be able to save and restore them in this way. To create the four DOSKEY macros described above and redefine the PgDn key as well, just run the batch file MACROTIP.BAT shown in Figure 3 or download it from PC MagNet as MACROT.BAT. □

Share your latest DOS and system discoveries through User-to-User. See the "How to Contact Us" sidebar in the Solutions column.

MACROTIP.BAT
Complete Listing

```
@ECHO OFF
REM ==== The ^[ in the next two lines
REM ==== represents the escape character.
DOSKEY ESC=ECHO ^[[$*
DOSKEY PESC=ECHO ^[$* $G PRN
DOSKEY LP=ECHO $* $G PRN
REM ==== The ^L in the next line
REM ==== represents the Ctrl-L character
DOSKEY FF=ECHO ^L $G PRN
REM ==== The ^[ in the next line represents
REM ==== the escape character.
ECHO ^[[0;81;"DOSKEY /m¦sort¦more";13p
```

Figure 3: **This batch file creates four handy DOSKEY macros, and redefines the Home key so it will list the available DOSKEY macros.**